Date: 24/07/2024

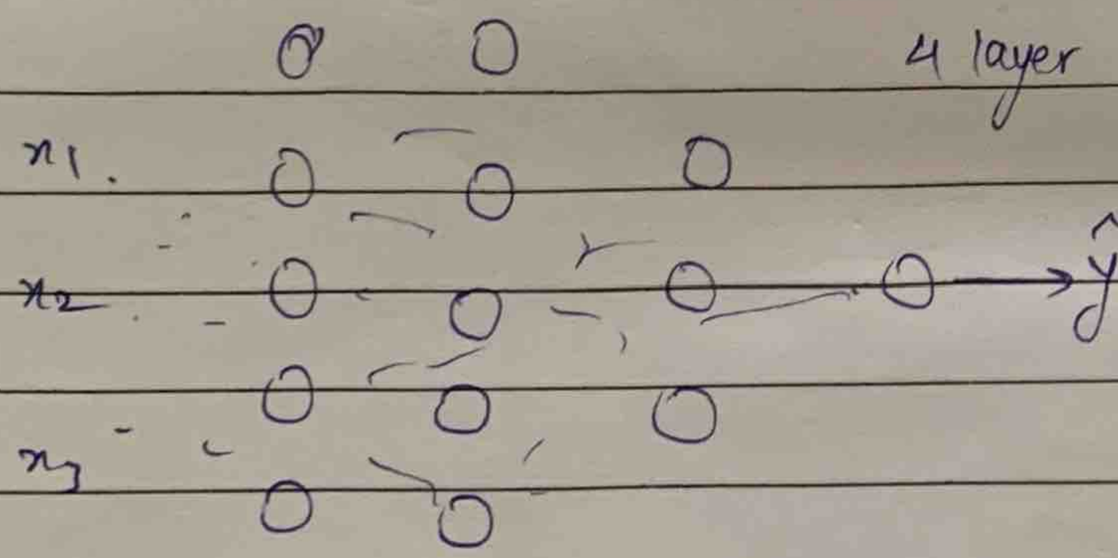
$\boxed{\text{WEEK 4}}$: Deep Neural Networks

Logistic Regression : "Shallow model"

5 hidden layers NN : "Deep model"

Deep NN notation :

4 layer NN



$L = 4$ (# layers)

$n^{[l]}$ = # no. of units in layer l.

$n^{[1]} = 5$, $n^{[2]} = 5$, $n^{[3]} = 3$, $n^{[4]} = 1$

$n^{[0]} = n_x = 3$

Date:

$a^{[l]}$ = activations in layer l

$a^{[l]} = g^{[l]}(z^{[l]})$

$w^{[l]}$ = weight for $z^{[l]}$

$x = a^{[0]}$ activations of layer 0 / input layer

$\hat{y} = a^{[L]}$ = activations of final layer

X : $z^{[1]} = w^{[1]} \overset{a^{[0]}}{x} + b^{[1]}$

$a^{[1]} = g^{[1]}(z^{[1]})$

$z^{[2]} = w^{[2]} a^{[1]} + b^{[2]}$

$a^{[2]} = g^{[2]}(z^{[2]})$

$\vdots$

$z^{[4]} = w^{[4]} a^{[3]} + b^{[4]}$; $a^{[4]} = g^{[4]}(z^{[4]})$

Date:

General form of forward propagation:

$$z^{[l]} = w^{[l]} a^{[l-1]} + b^{[l]}$$

$$a^{[l]} = g^{[l]}(z^{[l]})$$

Vectorised:

$$Z^{[1]} = W^{[1]} X + b^{[1]} \quad (X \to A^{[0]})$$

$$Z^{[2]} = \begin{bmatrix} | & | & | \\ z^{[2](1)} & z^{[2]2} & \cdots z^{[2](m)} \\ | & | & | \end{bmatrix}$$

$$A^{[1]} = g^{[1]}(Z^{[1]})$$

$$Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$$

$$A^{[2]} = g^{[2]}(Z^{[2]})$$

Over here 1 for loop will be used; ~~can't get rid of~~

for i (1 → L).

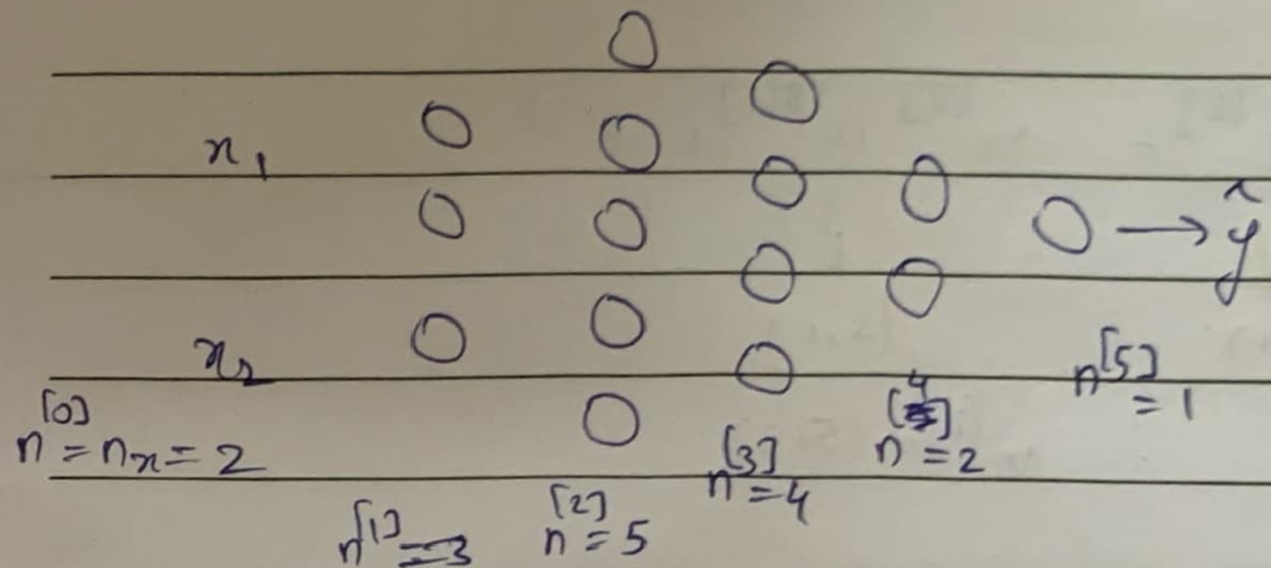
⋮

$$\hat{Y} = g^{[4]}(Z^{[4]}) = A^{[4]}$$

General form is same as above, just letters are now ~~capilasied~~ capitlaized.

Date:

Parameters of $W^{[l]}$ and $b^{[l]}$



$$Z^{[1]} = W^{[1]} x + b^{[1]} \qquad W^{[1]}: (n^{[1]}, n^{[0]})$$

$$\underset{(n^{[1]},1)}{\underset{(3,1)}{}} \quad \underset{(n^{[1]},n^{[0]})}{\underset{(3,2)}{}} \quad \underset{(n^{[0]},1)}{\underset{(2,1)}{}}$$

?

$$(n^{[1]}, 1) = (\quad) \cdot (n^{[0]}, 1)$$

$$(3,1) = (x, y) \cdot (2,1)$$

$$(x, y) = (3,2)$$

Date:

$W^{[2]} \div (5,3) \qquad (n^{[2]}, n^{[1]})$

$z^{[2]} = w^{[2]} a^{[1]} + b^{[2]}$

$\downarrow \qquad \downarrow$

$(5,1) \qquad (3,1)$

$\therefore w^{[2]} = (5,3)$

$W^{[3]} =: (4,5)$

$W^{[4]} : (2,4)$

$W^{[5]} : (1,2)$

General:- $W^{[l]} : (n^{[l]}, n^{[l-1]})$

$b^{[l]} : (n^{[l]}, 1)$

for dimensions $\begin{cases} dw^{[l]} \text{ same as } W^{[l]} \\ db^{[l]} \text{ sames as } b^{[l]} \end{cases}$

Date:

vectorised implementation :

$$z^{[1]} = W^{[1]} \, x + b^{[1]}$$

$$(n^{[1]}, 1) = (n^{[1]}, n^{[0]}) \cdot (n^{[0]}, 1) + (n^{[1]}, 1)$$

$$Z^{[1]} = W^{[1]} \, X + b^{[1]}$$

$$\begin{bmatrix} | & | & & | \\ z^{[1](1)} & z^{[1](2)} & \cdots & z^{[1](m)} \\ | & | & & | \end{bmatrix}$$

$$\therefore (n^{[1]}, m) = (n^{[1]}, n^{[0]}) \cdot (n^{[0]}, m) + \underbrace{(n^{[1]}, 1)}_{\text{broadcasting} \,\to\, (n^{[1]}, m)}$$

$z^{[l]}, a^{[l]} : (n^{[l]}, 1)$

$dZ^{[l]}, dA^{[l]} : (n^{[l]}, m)$

$Z^{[l]}, A^{[l]} : (n^{[l]}, m)$

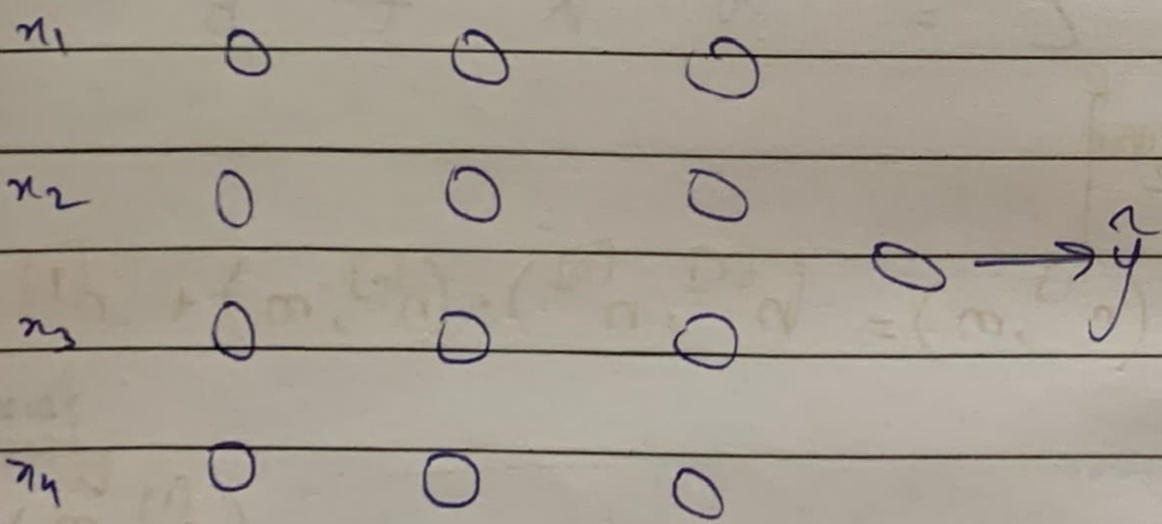
$l = 0 \quad A^{[0]} = X = (n^{[0]}, m)$

Date:

Circuit theory and Deep Learning:

There are functions with a "small" L-layer Deep NN than shallow NN using exponentially more hidden units to compute

Forward and Backward Functions:



Layer l: $W^{[l]}, b^{[l]}$

Forward: Input $a^{[l-1]}$, output $a^{[l]}$

$z^{[l]} = W^{[l]} a^{[l-1]} + b^{[l]}$ Cache $z^{[l]}$

$a^{[l]} = g^{[l]}(z^{[l]})$

Date:

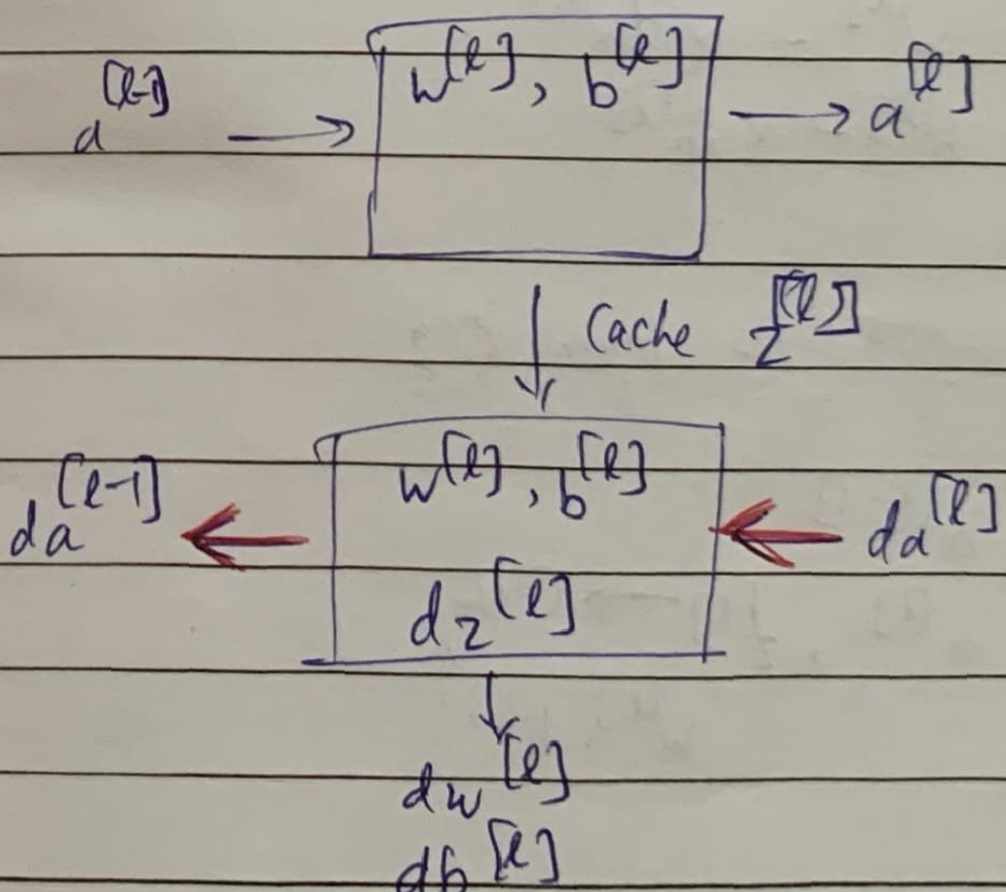
Backward:

Input $da^{[l]}$, cache $z^{[l]}$ output: $da^{[l-1]}$

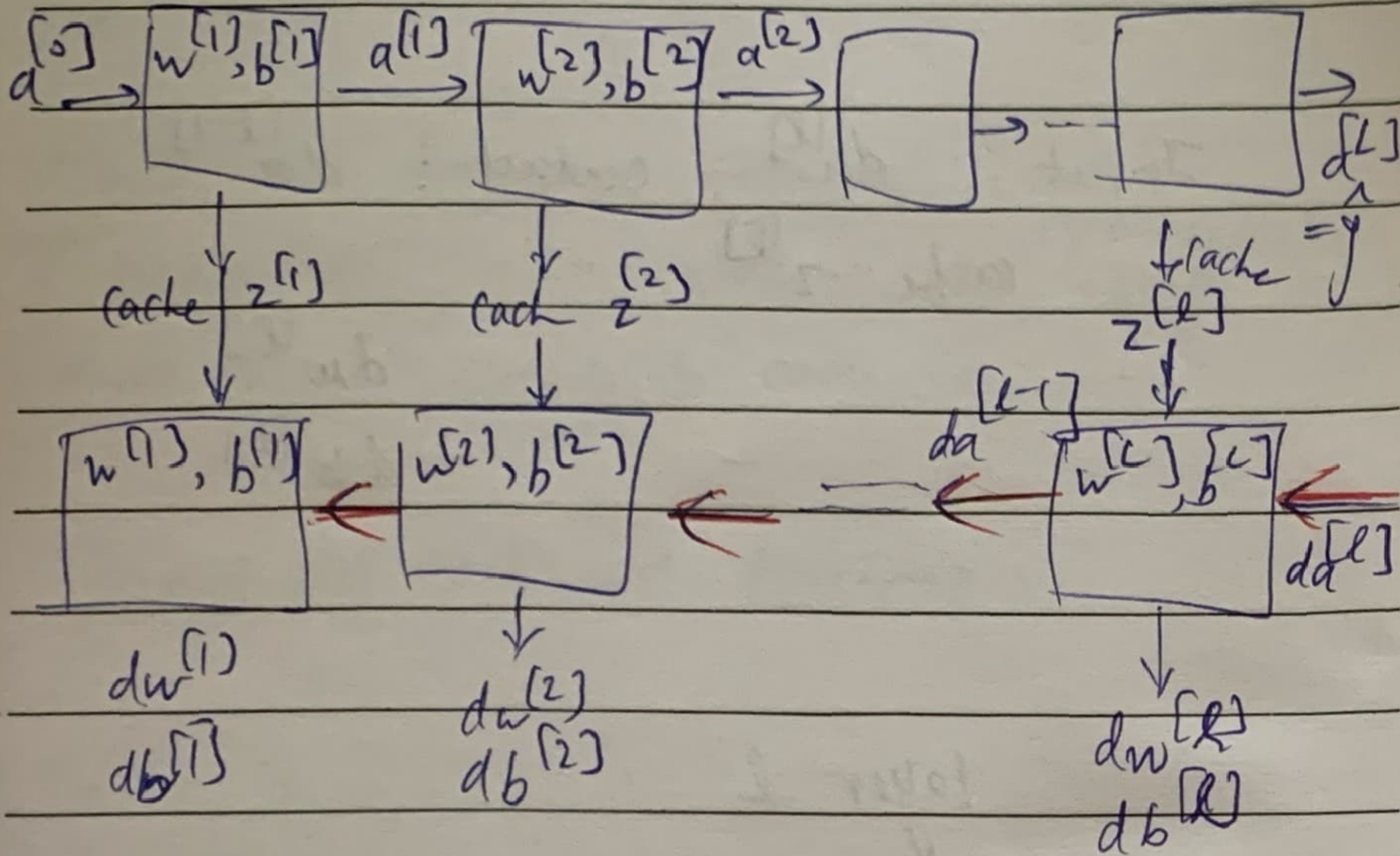
$dw^{[l]}$

$db^{[l]}$

layer l



Date:



$$w^{[l]} = w^{[l]} - \alpha\, dw^{[l]}$$
$$b^{[l]} = b^{[l]} - \alpha\, db^{[l]}$$

Forward propagation for layer $\{l\}$:

Input: $a^{[l-1]}$

output: $a^{[l]}, z^{[l]} \longrightarrow$ Cache

$w^{[l]}, b^{[l]} \uparrow$

Date:

$$z^{[l]} = w^{[l]} a^{[l-1]} + b^{[l]}$$

$$a^{[l]} = g^{[l]}(z^{[l]})$$

Vectorised :

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$$

$$A^{[l]} = g^{[l]}(Z^{[l]})$$

Backward propagation layer l:

Input : $da^{[l]}$

Output : $da^{[l-1]}, dw^{[l]}, db^{[l]}$

$$dz^{[l]} = da^{[l]} * g^{[l]'}(z^{[l]})$$

$$dw^{[l]} = dz^{[l]} \cdot a^{[l-1]T}$$

$$db^{[l]} = dz^{[l]}$$

$$da^{[l-1]} = W^{[l]T} \cdot dz^{[l]}$$

Date:

$$dz^{[l]} = W^{[l+1]T} \, dz^{[l+1]} * g^{[l]'}(z^{[l]})$$

Vectorised:

$$\boxed{dZ^{[l]} = dA^{[l]} * g^{[l]'}(Z^{[l]})}$$

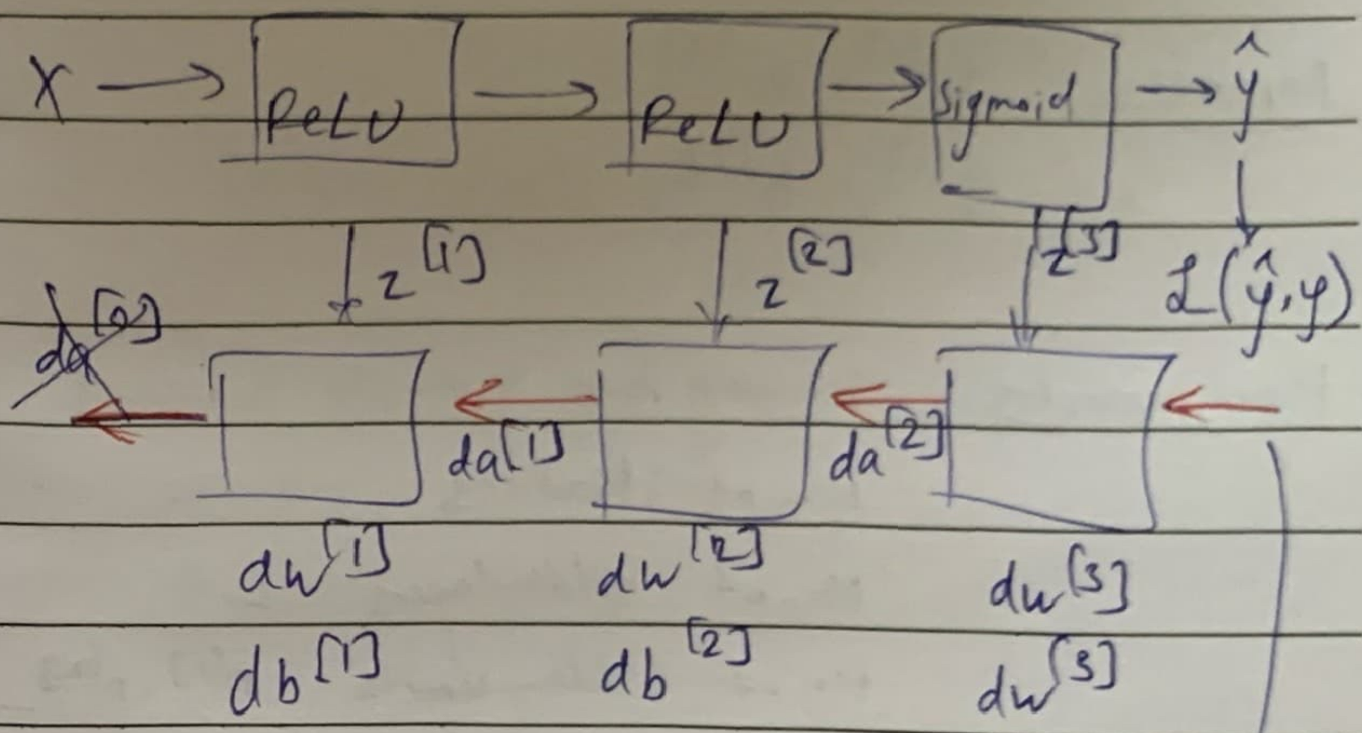
$$dW^{[l]} = \frac{1}{m} \, dZ^{[l]} \cdot A^{[l-1]T}$$

$$db^{[l]} = \frac{1}{m} \, np.sum(dZ^{[l]}, axis=1, keepdims=True)$$

$$dA^{[l-1]} = W^{[l]T} \cdot dZ^{[l]}$$

Date:

Summary



$$da^{[L]} = -\frac{y}{a} + \frac{(1-y)}{(1-a)}$$

Date:

What are hyperparameters:

Parameters: $w^{[1]}, w^{[2]} \ldots$
$b^{[1]}, b^{[2]} \ldots$

Hyper parameters: learning rate $= \alpha$
No. of iterations
No. of hidden layers L
No. of hidden units $n^{[1]}, n^{[2]} \ldots$
Choice of activation function.

Hyperparameters final decide the values of w & b.